Sending NxLogs with TLS in Nagios Log Server 2024R2

Purpose

This document describes how to setup encryption between Nagios Log Server and NXLog on Windows using self signed certificates. This document is intended for use by Nagios Log Server 2024R2 Administrators who would like encryption between NLS and their Windows NXLog clients. Encryption ensures that the traffic between the Windows machine and Nagios Log Server is not sent in plain text.

Overview

This documentation is broken up into the following sections:

- Create Certificates on the Nagios Log Server
  - Create a Certificate Authority (CA)
  - Create a certificate for the Nagios Log Server
  - Create a certificate for the Windows NXLog client
- Copy New Certificates
- Create Firewall Rule
- Create Input in Nagios Log Server using the certificates
- Configure NXLog to use the certificates

Prerequisites

It is assumed that you already have NXLog installed on your Windows machine. The installation steps and client are available in Nagios Log Server by clicking the green **+ Add Log Source** button at the top of the user interface. The following documentation is available as well:

Sending Windows Logs To Nagios Log Server

Terminology

For your information:

- **SSL**: Secure Sockets Layer
- **TLS**: Transport Layer Security

TLS replaces SSL, however the tools used to implement both generally use SSL in their name/directives. For simplicity reasons, the rest of this document will use the term SSL.

The steps in this documentation will create a CA and that CA will sign two certificates. This allows Nagios Log Server to use the CA to trust that the certificates used by the source and destination are valid.

Global Config vs Per Instance

This documentation walks you through creating certificate files that will be used in the Logstash Input that is created.

If you define this Input in the **Global Config**, you will be required to place the certificate files on **ALL** of your Nagios Log Server instances. If you do not, the configuration will **NOT** be applied on the instances that do not have the certificate files. This means that the input configuration will never be updated on these instances.

If you do not wish to implement the certificates on each Nagios Log Server instance, you will need to create the Input as a **Per Instance** config for the instance that has the certificate files (this will be explained later).

Installing Necessary Components

Establish a terminal session to your Nagios Log Server and as root and execute the following command:

RHEL | CentOS | Oracle Linux

```
yum install -y mod_ssl openssl
```

Debian | Ubuntu

```
apt-get install -y openssl
```

All of the remaining steps will be performed from within the root user's home directory to ensure the files you create are not accessible to anyone except the root user. Change into the home directory with this command:

```
cd ~
```

You will continue to use this terminal session throughout this documentation.

Create a Certificate Authority

1. Generate the Private Key for your Nagios Log Server instance(s) by running the following command:

   ```
   openssl genrsa -out ca.key 2048
   ```

2. Generate a Certificate Signing Request (CSR) and sign the key by running:

   ```
   openssl req -x509 -new -nodes -key ca.key -sha256 -days 1024 -out ca.pem
   ```

   a. During the execution, you'll be prompted to enter certain details. Some fields can be left blank. Below is an example of the expected input:

   ```
   Country Name (2 letter code) [XX]:AU
   State or Province Name (full name) []:NSW
   Locality Name (eg, city) [Default City]:Sydney
   Organization Name (eg, company) [Default Company Ltd]:My Company Pty Ltd
   Organizational Unit Name (eg, section) []:
   Common Name (eg, your name or your server's hostname) []:ca
   Email Address [] :
   ```

   Note: The `Organizational Unit Name` and `email address` fields are optional and can be left blank.

Create a Nagios Log Server Certificate:

1. Generate the Private Key for your Nagios Log Server instance(s) by running the following command:

   ```
   openssl genrsa -out device-nls.key 2048
   ```

2. Generate a Certificate Signing Request:

   ```
   openssl req -new -key device-nls.key -out device-nls.csr
   ```

a. During the execution, you'll be prompted to enter certain details. Some fields can be left blank. Below is an example of the expected input:

```
Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:NSW
Locality Name (eg, city) [Default City]:Sydney
Organization Name (eg, company) [Default Company Ltd]:My Company Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:nls
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Note: The `Organizational Unit Name`, `email address`, and `company name` are optional and can be left blank. The `challenge password` is not required.

3. Sign the Certificate using the following command:

   Note: the following is one long command that has wrapped over two lines:

   ```
   openssl x509 -req -in device-nls.csr -CA ca.pem -CAkey ca.key -CAcreateserial -out device-nls.crt \
   -days 500 -sha256
   ```

   Which should produce output saying: `Certificate request self-signature ok.`

Create Windows NXLog Certificate

1. Generate a private key for your Windows NXLog client by running:

   ```
   openssl genrsa -out device-nxlog.key 2048
   ```

2. Generate the Certificate Signing Request (CSR):

   ```
   openssl req -new -key device-nxlog.key -out device-nxlog.csr
   ```

a. During the execution, you'll be prompted to enter certain details. Some fields can be left blank. Below is an example of the expected input:

```
Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:NSW
Locality Name (eg, city) [Default City]:Sydney
Organization Name (eg, company) [Default Company Ltd]:My Company Pty Ltd
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:nxlog
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Note: The `Organizational Unit Name`, `email address`, and `company name` are optional and can be left blank. The `challenge password` is not required.

3. Sign the Certificate using the following command:

   Note: The following is one long command that has wrapped over two lines:

```
openssl x509 -req -in device-nxlog.csr -CA ca.pem -CAkey ca.key -CAcreateserial -out device-nxlog.crt \
-days 500 -sha256
```

Which should produce output saying: `Certificate request self-signature ok.`

Copy New Certificates

Use the following commands to copy the new certificates to the correct locations:

```
cp ca.key /etc/pki/tls/private/
cp device-nls.key /etc/pki/tls/private/
cp ca.pem /etc/pki/tls/certs/
cp device-nls.crt /etc/pki/tls/certs/
```

If you plan on creating the Input as part of the Global Config, you will need to copy these certificate files to all the instances in your Nagios Log Server cluster. Please refer to the Global Config vs Per Instance section of this document for more information.

Create Firewall Rule

You need to create a firewall rule to allow the incoming TCP traffic. In your terminal session execute the following commands (depending on your operating system version):

RHEL | CentOS | CentOS Stream | Oracle Linux

```
firewall-cmd --zone=public --add-port=7777/tcp
firewall-cmd --zone=public --add-port=7777/tcp --permanent
```

Debian:

The local firewall is not enabled on Debian by default and no steps are required here. **IF** it is enabled then the commands are:

```
iptables -I INPUT -p tcp --destination-port 7777 -j ACCEPT
```

Ubuntu:

The local firewall is not enabled on Ubuntu by default and no steps are required here. **IF** it is enabled then the commands are:

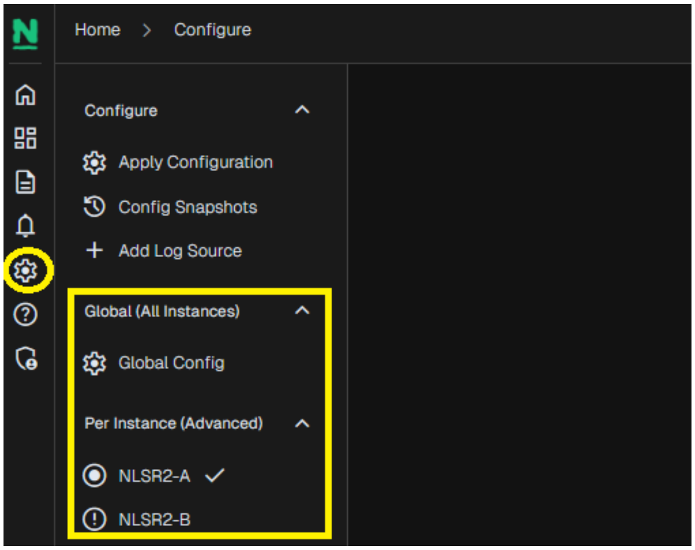
```
sudo ufw allow 7777/tcp
sudo ufw reload
```

If you plan on creating the Input as part of the Global Config, you will need to create this firewall rule on all the instances in your Nagios Log Server cluster.

Create Input

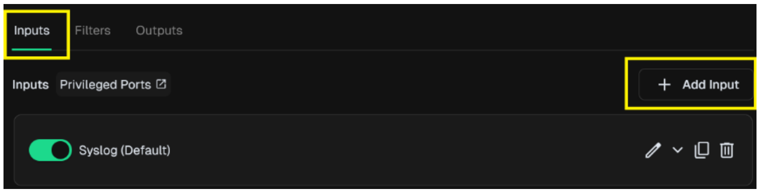
Now we will create Input that uses the certificates you have created and will be listening on TCP port 7777.

1. Login to one of your Nagios Log Server instances as an Admin user and the **Configure** (gear) icon in the lefthand menu:
2. Select one of these options:
    a. For a **Global Config**: In the left pane under **Global (All Instances)** click **Global Config**
    b. For a **Per Instance Config**: In the left pane under **Per Instance (Advanced)** click the **Log Server Instance** which has the certificate files you created.



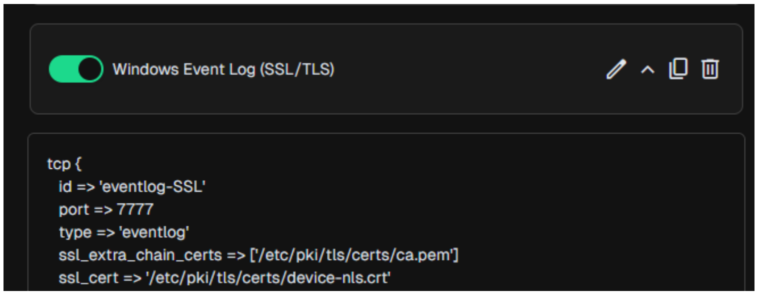
Note: The remaining steps are common to either option.

3. On the right side of the screen click the **+ Add Input** button. A new block appears at the bottom right of the Inputs table.



4. Type a unique **Block Name** for the input which will be: **Windows Event Log (SSL/TLS)**

5. In the text area field enter the following code (you can copy and paste):



Windows Event Log (SSL/TLS)

```
tcp {
  id => 'eventlog-SSL'
  port => 7777
  type => 'eventlog'
  ssl_extra_chain_certs => ['/etc/pki/tls/certs/ca.pem']
  ssl_cert => '/etc/pki/tls/certs/device-nls.crt'
```

```
tcp {
   id => 'eventlog-SSL'
   port => 7777
   type => 'eventlog'
   ssl_extra_chain_certs => ['/etc/pki/tls/certs/ca.pem']
   ssl_cert => '/etc/pki/tls/certs/device-nls.crt'
   ssl_key => '/etc/pki/tls/private/device-nls.key'
   ssl_enabled => true
   ssl_verify => false
   codec => json  {
        charset => 'CP1252'
   }
}
```

6. Finally, Click the **Save & Apply** button to create this filter and apply the configuration.

Configuring NXLog on Windows

Follow these steps to setup NXLog on your Windows.

Copy the Certificate to Windows

You need to transfer the CA certificate and the NXLog certificate to your Windows machine. NXLog may be installed in different locations depending on the version. For older versions, it may be in `C:\Program Files(x86)`, so verify the installation path before proceeding.

1. Copy the following files from your Linux system to Windows:
    a. CA Certificate
    Source: `/root/ca.pem`
    Destination: `C:\Program Files\nxlog\cert\ca.pem`
    b. NXLog Certificate
    Source: `/root/device-nxlog.crt`
    Destination : `C:\Program Files\nxlog\cert\device-nxlog.crt`

Note: You can transfer the files using:

- WinSCP
- SCP in Powershell (available in Windows 11)
- Manual Copy-Paste: Open the certificate files on Linux using `cat /root/ca.pem`, copy their contents and paste them into Notepad on Windows, and save them with the correct filenames.

Update the NXLog Configuration

2. Open a Notepad as an administrator
3. Open the NXLog configuration file: `C:\Program Files\nxlog\conf\nxlog.conf`
4. Find the section that looks like this and comment it out by adding `#` at the beginning of each line:

```
#<Output out>
#   Module om_tcp
#   Host 10.25.5.99
#   Port 3515
#
#   Exec  $tmpmessage = $Message; delete($Message);
rename_field("tmpmessage","message");
#   Exec  $raw_event = to_json();
#
#  # Uncomment for debug output
#  # Exec file_write('%ROOT%\data\nxlog_output.log', $raw_event + "\n");
#</Output>
```

Note: In the example above, the `Host 10.25.5.99` line contains the IP Address of your Nagios Log Server instance. You'll need this IP for the next step.

5. Change the `Host` to the IP address of your Log Server instance.
6. Change `Exec $source_ip` to the IP Address of your Windows machine.

Example:

```
<Output out>
   Module         om_ssl
   Host           10.25.5.99
   Port           7777
   CertFile       C:\Program Files\nxlog\cert\device-nxlog.crt
   CAFile         C:\Program Files\nxlog\cert\ca.pem
   AllowUntrusted TRUE
   Exec $tmpmessage = $Message; delete($Message);rename_field("tmpmessage","message");
   Exec $raw_event = to_json();
   Exec $raw_event = $raw_event + $source_ip;
   Exec $source_ip = '192.168.100.150';
</Output>
```

7. After making these changes, **Save** the file and close Notepad.

Restart the nxLog Service

8. Restart the nxlog service on the Windows machine using the following commands in a Command Prompt (Run as Administrator):

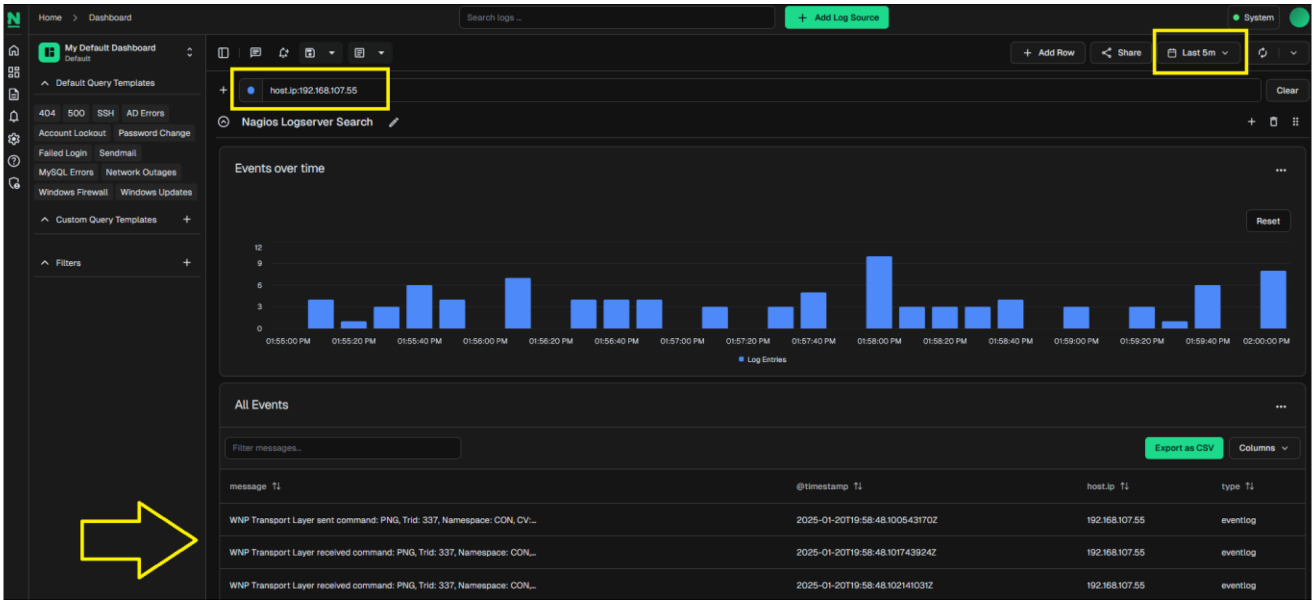
```
sc stop nxlog
sc start nxlog
```

Verify Incoming Logs

To confirm that Nagios Log Server is receiving data from the Windows server navigate to the **Dashboards** page. Perform a **Query** on the host field using the **IP Address** of your **Windows** host, and choose ‘Last 5m’ from the timepicker dropdown:

```
host.ip:<Windows Host Address>
```

Here is an example that shows the received logs appearing in the **All Events** panel:



host.ip:192.168.107.55

message	@timestamp	host.ip	type
WNP Transport Layer sent command: PNG, Trid: 337, Namespace: CON, CV:...	2025-01-20T19:58:48.100543170Z	192.168.107.55	eventlog
WNP Transport Layer received command: PNG, Trid: 337, Namespace: CON,...	2025-01-20T19:58:48.101743924Z	192.168.107.55	eventlog
WNP Transport Layer received command: PNG, Trid: 337, Namespace: CON,...	2025-01-20T19:58:48.102141031Z	192.168.107.55	eventlog

Additional Information

If you would like to verify that traffic is encrypted, you can verify this by using tcpdump. First you must have tcpdump installed on your Nagios Log Server which can be done with this command:

RHEL | CentOS | Oracle Linux

```
yum install -y tcpdump
```

Debian | Ubuntu

```
apt-get install -y tcpdump
```

Once installed execute the following command to observe the traffic:

```
tcpdump -i ens32 -nnvXSs 0 host 10.25.14.91
```

In that command, `ens32` is the network interface on the Nagios Log Server and `10.25.14.91` is the IP address of the Windows machine.

Here is example output **before** implementing SSL/TLS:

```
11:40:48.857072 IP (tos 0x0, ttl 128, id 31372, offset 0, flags [DF], proto
TCP (6), length 204)
    10.25.14.91.61978 > 10.25.5.99.3515: Flags [P.], cksum 0xc5dc (correct),
seq 1015102624:1015102788, ack 1368467930, win 16425, length 164
   0x0000:  4500 00cc 7a8c 4000 8006 57c8 0a19 0e51  E...z.@...W....Q
   0x0010:  0a19 0555 f21a 0dbb 3c81 3ca0 5191 29da  ...U....<.<.Q.).
   0x0020:  5018 4029 c5dc 0000 7b22 4576 656e 7452  P.@)....{"EventR
   0x0030:  6563 6569 7665 6454 696d 6522 3a22 3230  eceivedTime":"20
   0x0040:  3137 2d30 342d 3138 2031 313a 3430 3a34  17-04-18.11:40:4
   0x0050:  3422 2c22 536f 7572 6365 4d6f 6475 6c65  4","SourceModule
```

You can see in the right-hand side the plain text such as “`EventReceivedTime":"2017-04-18.11:40:44`”

Here is an example output **after** implementing SSL/TLS; you can see on the right hand side the data in encrypted and cannot be understood:

```
11:47:07.228206 IP (tos 0x0, ttl 128, id 1497, offset 0, flags [DF], proto
TCP (6), length 274)
    10.25.14.91.54713 > 10.25.5.99.7777: Flags [P.], cksum 0x3ac9 (correct),
seq 4122608981:4122609215, ack 932892309, win 16074, length 234
   0x0000:  4500 0112 05d9 4000 8006 cc35 0a19 0e51  E.....@....5...Q
   0x0010:  0a19 0555 d5b9 1e61 f5ba 0555 379a ce95  ...U...a...U7...
   0x0020:  5018 3eca 3ac9 0000 1703 0100 204b f03e  P.>.:........K.>
   0x0030:  a312 5aa3 efc2 3cea 5830 4c8c 2983 f47a  ..Z...<.X0L.)..z
   0x0040:  dc67 3524 7961 dfb4 73de c64e b517 0301  .g5$ya..s..N....
   0x0050:  00c0 459c 61e3 b309 b963 b3ab 599c 0b55  ..E.a....c..Y..U
   0x0060:  221b c8dd 41e7 ffac 1b7a 6ba9 b5df 0dc5  "...A....zk.....
   0x0070:  b902 827c 8076 5b83 7f6c 79f8 e57c ea6c  ...|.v[..ly..|.l
   0x0080:  b628 e274 aa64 1b58 3348 39c2 856d 79ab  .(.t.d.X3H9..my.
   0X0090:  1cc7 a825 016d 5b96 e155 4f6c 2b69 4fae  ...%.m[..UOl+iO.
   0X00a0:  3704 d9f3 6302 39a9 fd4c 5020 839b 324f  7...c.9..LP...2O
```

You can see in the right-hand side the data is encrypted and cannot be understood.

Finishing Up

This completes the documentation on Sending NxLogs with SSL/TLS in Nagios Log Server 2024R2. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum   Visit Nagios Knowledge Base   Visit Nagios Library